COMP 4000A/5102F: Distributed Operating Systems
Fall 2019 Midterm Exam

October 16, 2019

Instructor: Anil Somayaji
80 minutes, Open Book

Answer two out of the following three questions. If you answer more than two questions, clearly indicate which ones should be graded. Please write your answers in a separate exam booklet, or, alternately type them on a computer and submit them via cuLearn or email them to `anil.somayaji@carleton.ca`.

The exam is open book, open note, open Internet. The only thing you may not do is discuss questions with other individuals. In other words, no emailing/IM/texting/whatever with other people during the exam!

How to Answer the Questions: Answer each question with a small essay. When the question has multiple parts (e.g., asking for you to discuss two or three separate systems), please do not answer them separately; instead, use them to help structure your small essay answer. Be specific but also make appropriate generalizations. Be concise—the essays will be long enough if you really answer the questions and time is short.

Show me what you understand, not what you remember.

Good luck!

1. Argue for or against: The design goals of the early Internet (as discussed in "Resource-Sharing Computer Communications Networks") were also the key goals in distributed operating systems research and development in the 1980's and early 1990s. Be sure to discuss using multiple examples.

2. Remote procedure calls (RPC) and distributed shared memory (DSM) are both approaches to making what would normally be a local operation work in a distributed context. With RPC, the focus is on the function call, while with DSM the focus is on arbitrary memory access. Compare and contrast RPC and DSM as approaches to achieve network transparency. Specifically, what are their respective implementation requirements, what are the potential benefits, and what are the potential problems? Explain using at least three specific examples.

3. Distributed operating system mechanisms often violate UNIX semantics even when attempting to mimic them. What kinds of UNIX semantics are preserved? What are broken? Why do you think designers made these choices? Discuss using at least three examples.